

Workshop

Fakta

Kurs:	Systemutvecklare C/C++ Extended 2024
Klass:	SUVx24
Teknikområde:	Programspråket C, in- och utmatning, funktioner

Learning Target

Programmering i C och C++

Grundläggande begrepp inom programmering.

Skapa källkod med C syntax. Tema funktioner.

Grundläggande in- och utmatning

In- och utmatning av resultat från ett program med hjälp av standardfunktioner i C.

Användning av utvecklingsmiljöer

Använda *Visual Studio Code* (förkortat VSC) för att skapa källkod, kompilera och exekvera program skapat med programspråket C.

Grundläggande felsökning

Grundläggande tekniker för felsökning och problemlösning i C

Introduktion

Programspråket C ger oss möjlighet att kunna manipulera arbetsminnet i datorn. Det för med sig vissa risker också, men anses vara en styrka med språket, som t ex *Java* och *Python* inte kan. Sättet vi kan komma åt specifika minnesadresser på, är genom att använda pekare (pointer).

När vi skapar kod är det ganska vanligt att det smyger sig in fel medan vi kodar. Fel i kod kallas en bug och vissa fel är väldigt uppenbara (läs *syntaxfel* eller *formella fel*) och fångas upp av kompilatorn, som berättar vad problemet är. Men ibland gör vi tankevrpor och skapar *logiska fel*. Då kan vi behöva ett verktyg som hjälper oss i vårt detektivarbete att finna felen. Ett sådant verktyg brukar kallas *debugger*, vilket ofta finns för den kompilator du valt. Vi ska titta på vad man kan göra i *VS Code*.

Workshop

Förberedelse

Skapa en ny projektmapp (katalog) i den projektmapp du använder för kodning, typ:

```
C:\Chas\SUVx24\Workshop10\
```

Starta VSC på det sätt du tycker är lättast och öppna den nya projektmappen.

Innehåll

Träna på att använda pekare i C

Uppgift 1

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop11-1.c**

Skapa ett program som

- Deklarerar fyra pekarvariabler.
 - En för varje datatyp `int`, `float`, `double` och `char`.
- Använd funktionen `sizeof()` och skriv ut storleken i bytes för respektive variabel.
 - Skriv ut på formen:
`Int size = ?`
 - där frågetecknet ersätts med returvärdet från `sizeof()`-funktionen.
- Kör programmet.
- Vad visar resultatet?
 - Kan du förklara varför?

Workshop

Uppgift 2

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop11-2.c**

Använd följande kod:

```
#include <stdio.h>

int main() {

    int test;
    int *pInt = 0;
    test = 10;

    // Print the value of test variable
    printf("\nTest's value = %i", );
    // Print the address of test variable
    printf("\nTest's address as integer = %", ); // Which format
    specifier to use?
    // Print the address of test variable in hexadecimal format
    printf("\nTest's address in hexadecimal format = %", ); // Which
    format specifier to use?

    pInt = &test;
    // Print the value of pInt variable

    // Print the address of pInt variable

    // Print the address of pInt variable in hexadecimal format

    // Replace the value and print the value of test variable
    *pInt = 15;
    printf("\nNow, test's value = %i", );

    return 0;
}
```

- Fyll i de tre första `printf()`-funktionerna med rätt variabel och eventuellt tillhörande symbol för att skriva ut rätt sak. Välj också matchande identifierare för formatet på variabelns utskrift.
- Komplettera med tre nya `printf()` under respektive kommentar efter tilldelningen till `pInt`.
- Fyll även i den sista `printf()` med rätt variabel för att se effekten av pekarens tilldelning.

Workshop

Uppgift 3

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop11-3.c**

Använd följande kod:

```
#include <stdio.h>

int main() {

    int e[3] = {1, 12, 123};
    int i, *pInd;

    pInd = &i;

    for(i = 0; i < 3; i++)
    {
        printf("\nr[%i] address = %u %u %u %u %u %u ", i, &e[i], &i[e],
pInd + i, i + pInd, e + i, i + e);
        printf("\nr[%i] value = %u %u %u %u %u %u ", i, &e[i], &i[e],
*(pInd + i), *(i + pInd), *(e + i), *(i + e));
    }

    return 0;
}
```

Kör programmet och studera resultatet och jämför med koden.

- Vad drar du för slutsatser kring operationerna mellan variablerna?